1 Oppgave 1

Gjør kjøretidsanalyse av følgende algoritmer

1.1 a)

```
public static double F1(double x, int k) {
    if (k == 1) {
        return x;
    }
    return x * F1(x, k - 1);
```

Svar. `if` - setningen er en operasjon, og har kjøretid: $O(1)$. `return x * F1(x, k - 1)` teller for tre operasjoner, og har kjøretid $O(3)$. Disse operasjonene kjøres `k` ganger. Siste operasjon: `return x` kjøres en gang, og teller for to operasjoner, og har kjøretid $O(2)$. Vi ender opp med en eksakt kjøretid $O(4k + 2)$. Dette er i Big-O notasjon lik $O(k)$.

1.2 b)

```
public static double F2(double x, int k) {
    if (k == 1) {
        return x;
    }
    int k2 = k/2;
    return F2(x, k2) * F2(x, k2);
```

Svar. Operasjonene har følgende kjøretider:

Operasjon	Kjøretid
`if (k == 1)`	1
`return x;`	2
`int k2 = k / 2;`	2
`retun F2(x, k2) * F2(x,k2)`	4

Vi har at `F2(x, 1) = 2`, og ettersom `k2` halveres i hvert funksjonskall, og `return` setningen kjører `F2` to ganger, ender vi opp med $O(7k + 2)$, som er $O(k)$ i Big-O notasjon

1.3 c)

```
public static double F3(double x, int k) {
    if (k == 1) {
        return x;
    }
    double res = F3(x, k/2);
    return res*res;
}
```

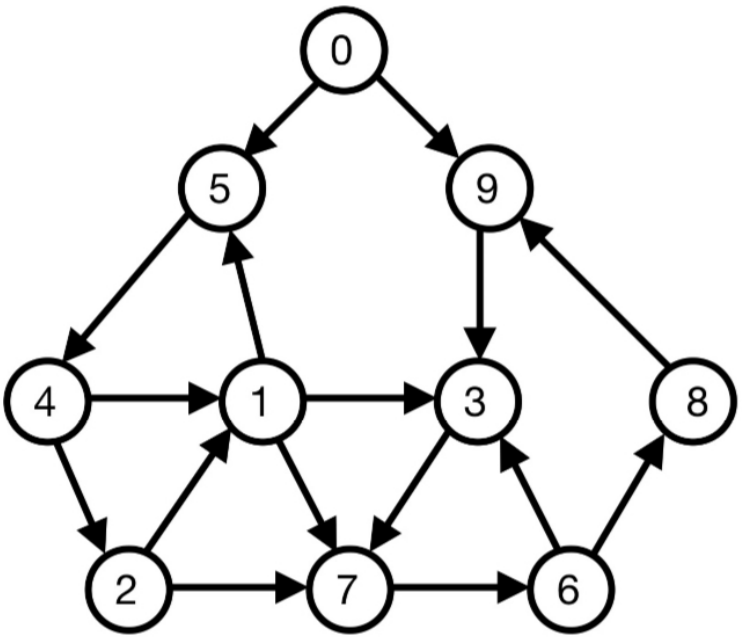
Svar. Operasjonene har følgende kjøretider:

Operasjon	Kjøretid
`if (k == 1)`	1
`return x;`	2
`double res = F3(x, k / 2)`	3
`return res*res`	2

Vi har at `F3(x, 1)` har kjøretid $O(2)$. Videre har vi `F3(x, k)` har kjøretid $O(5 \log k)$, vi får Big-O kjøretid på $O(\log k)$.

2 Oppgave 2

Gi rekkefølgen på noder besøkt for følgende graf



2.1 a)

Svar. DFS

$$0, 5, 4, 1, 3, 7, 6, 8, 9, 2$$

2.2 b)

Svar. Rekkefølgen på nodene når DFS er ferdig med å behandle nodene.

$$9, 8, 6, 7, 3, 1, 2, 4, 5, 0$$

2.3 c)

Svar. Gi rekkefølgen på de sterkt sammenhengende komponentene som de oppdages med Kosaraju-Sharir algoritmen.

$$(9, 3, 7, 6, 8)(1, 5, 4, 2)(0)$$

2.4 d)

Svar. Gi rekkefølgen på nodene etter første besøk iht. BFS

$$0, 5, 9, 4, 3, 1, 2, 7, 6, 8$$

3 Oppgave 3

3.1 a)

Hva er kjøretiden til quicksort dersom vi alltid velger det første elementet som pivot og input er sortert i stigende rekkefølge ved start?

Svar.

$$O(n^2)$$

3.2 b)

Hva er kjøretiden til innsettingssortering dersom input er sortert i synkende rekkefølge ved start?

Svar.

$$O(n^2)$$

3.3 c)

Hva er kjøretiden til mergesort dersom input er sortert i stigende rekkefølge ved start?

Svar.

$$O(n \log n)$$

3.4 d)

Hva er kjøretiden til mergesort dersom input er sortert i stigende rekkefølge ved start og vi hopper over alle "merge" operasjoner av to lister A og B hvis siste elementet i A er mindre eller lik første elementet i B?

Svar.

$$O(n)$$